Concept Formation vs. Logistic Regression: Predicting Death in Trauma Patients

Mirsad Hadzikadic, Ph.D., Carolinas Medical Center/University of North Carolina
Anne Hakenewerth, Carolinas Medical Center
Ben Bohren, Carolinas Medical Center
Jim Norton, Ph.D., Carolinas Medical Center
Binita Mehta, Carolinas Medical Center
Cindy Andrews, Carolinas Medical Center
Charlotte, North Carolina

This paper discusses two classification models, one based on concept formation and the other using standard logistic regression. The models are first explained in some detail and then evaluated on the same population of trauma patients. The goal of both systems is to predict the outcome of those patients. The results are summarized and explained in terms of differing algorithms of the two models.

INTRODUCTION

The purpose of this study was to compare two models for predicting death in trauma patients, one based on logistic regression and the other utilizing concept formation, an artificial intelligence approach to classification. While logistic regression is an accepted standard for quantifying classification "power" of a given data set, concept formation is a novel method which provides more flexibility in dealing with data sets containing incorrect or missing data as well as a large number of variables.

By comparing the two methods the authors have hoped to better understand the elements of trauma that contribute the most to death of affected patients. If successful, this study may eventually lead to suggestions that will both improve the quality and reduce the cost of care of trauma patients by promoting the procedures that address the most critical aspects of the patients injury.

The remainder of this paper will provide a brief description of the two models, followed by an explanation of obtained results and their interpretation.

CONCEPT FORMATION

Concept formation is a machine learning technique for summarizing known instances in the form of tree/hierarchy. An instance is defined as any object, event, or place that can be described in terms of attribute/value pairs. Concepts are then represented as intermediate nodes whose description provides a summary of all instances stored in the corresponding subtrees. Ideally, all instances which lead to the same consequence would be grouped in the same branch of the tree. After a tree is obtained, it can then be used to predict the classification of new instances.

In order to be effective, concept formation systems need to be unsupervised and incremental. The term *unsupervised* indicates that there is no teacher to decide on either the number or identity of the concepts to be learned by the system. Consequently, the system is provided with examples of the classes to be formed, but without any indication as to which class those examples belong to. The system is expected to uncover not only the classes themselves, but their description and subclass structure as well. Such an approach is well suited for domains which are not completely understood due to their complexity and/or incomplete information. Medicine is an example of a domain where much remains to be explained despite the fact that a tremendous effort already has been invested in it.

The term *incremental*, on the other hand, indicates that examples/instances are acquired one at a time. While it

0195-4210/95/$5.00

is certainly possible to form concepts by looking at all instances at once, such an approach would not be advisable in the context of medical decision making where each medical practitioner can build his/her own decision tree, i.e. a hierarchy of categories, from his/her own previous cases. This patient base keeps growing over time and it would be impractical to store a full description of all instances. The system should rather keep a description of all intermediate nodes and some informative (typical and not so typical) cases. Also, its real-time use suggests that the system should incorporate new cases into an existing tree rather than regenerate the whole tree from scratch.

For the purpose of this study we used INC2.5, a system developed by Hadzikadic and his collaborators (Hadzikadic, 1991, 1992a, 1992b; Hadzikadic and Yun, 1989; Bohren and Hadzikadic, 1994). INC2.5 is an example of an incremental concept formation tool. A node in the INC2.5 tree has a description consisting of the following data: *name of the node, list of attributes and associated values, measure of cohesiveness, number of children, and number of instances stored under this class.* Each node has an identical list of attributes. For each attribute the node will store all values found in its instance descriptions. A node containing a single instance will have *zero* or more values associated with each attribute. A class node is identical in structure to an instance node, but contains the union of all instance values stored under it. When a value occurs more than once, the number of occurrences of the attribute value in that branch of the tree is displayed in the attribute list. Upon calculating the similarity between two instances, the two attribute lists are compared and return a number reflecting both distinctive and common features.

The main components of INC2.5 are the class-membership evaluation function and the tree-searching algorithms. The evaluation function can be broken into two components, *similarity* and *cohesiveness*. As shown in Equation 1, the similarity of two nodes is based on the comparison between the two sets of attribute/value pairs, where **c(A,B)** represents the contribution of the common features of **a** and **b**; **d(A,B)** introduces the influence of the features of **a** not shared by **b**. The function is derived from the *contrast model* (Tversky, 1977), which defines similarity as a linear combination of common and distinctive attribute/value pairs, *features*.

$$s(A,B) = \frac{sim(A,B) + sim(B,A)}{2}$$

$$sim(A,B) = \frac{c(A,B) - d(A,B)}{c(A,B) + d(A,B)} \qquad \text{(Eq 1)}$$

$$cm(a) = \frac{sp(a)}{\binom{|a|}{2}} \qquad \text{(Eq 2)}$$

$$sp(a) = sp(i) + \sum_{J,k} (J,K) \times |j| \times |k| \therefore (i,j,k \in G_a)$$

Cohesiveness measures the average similarity of all pairs of instances contained in a class. Equation 2 is the formula used to calculate cohesiveness, where G_a represents the set of all children of **a** and **|a|** equals the number of instances stored under **a**. It ranges from 1.0 to -1.0. Cohesiveness reflects the similarity between all instances under a given node. In the special case where the node is an instance, singleton, the value of the cohesiveness is irrelevant. However, for INC2.5's evaluation function a singleton is assumed to have a cohesiveness of 0.0. A class node will have a cohesiveness measure of 1.0 if and only if all of its children are identical. On the other hand, a node in which the children are completely opposite would have a cohesiveness measure of -1.0.

INC2.5 uses six operators during the tree building process: create, extend, merge, delete, pull-in, and pull-out. Create forms a new class for an instance found to be dissimilar to all examined classes, while extend adds a new instance to the most similar class found. Merge and delete combine and divide classes based on the effect of a new instance. For example, if a new instance is similar to half of the children in a class then a new class will be formed by merging the most similar children and recursively classifying the new instance into the new class. The *most similar children* are defined here as

those children whose degree of similarity with the new instance is within a window W of the most similar child. The default value of W is set to 75%. The delete operator is used to undo the consequences of an unsuccessful merge operation. Pull-in and pull-out operators are also introduced to help correct previous misclassifications, although at the finer level of detail than the delete operator. They are applied when the system is updating the path from the newly placed instance to the root.

LOGISTIC REGRESSION

Logistic Regression is a statistical model that allows for a quantitative relationship for a dichotomous event that depends upon several independent variables. Cornfield (1962) proposed this model in predicting coronary heart disease. The outcome (dependent) variable must have only two choices, e.g. occurs or not, alive or dead, etc. The independent variables can be either on the interval or nominal scale. This mathematical model predicts the probability of outcome of an event p with k independent variables $x_1, x_2, ..., x_k$ according to the following equation:

$$p(event|X) = \frac{1}{1+e^{-(Bo+B1\ x1+B2\ x2+\ldots+Bk\ xk)}}$$

Any nominal scale variable must first be re-parameterized so that an individual variable x_i is either on the interval scale or dichotomous.

A brief description will be given of how the coefficients B_i are calculated. For more details see Cox (1970) or Agresti (1990). The method used is a standard one that is used in many statistical models. It is known as the method of maximum probability estimation. If the event is survival then the probability that the *i*th individual survives is

$$pi(event|Xi) = \frac{1}{1+e^{-(Bo+B1\ x1i+B2\ x2i+\ldots+Bk\ xki)}}$$

and the probability that the *i*th individual dies is just 1-p_i.

The probability function is simply the product of the individual p_is for those individuals that survive multiplied by those q_is for those individuals that die. The assumption of independence, i.e. the occurrence of the event in one person is completely independent from the occurrence of an event for another individual, allows one to multiply the individual probabilities. With this method, one would like to choose the values of the B coefficients so as to make the probability function a maximum for this particular sample of people. From calculus, the maximum values for a set of equations with unknowns is found by taking the partial derivatives and setting them equal to zero. Since the logarithm is a monotonically increasing function, the maximums for the logarithm of a function will be the same as for the original function. In this case it is easier to calculate the derivatives for the natural logarithm of the probability function and then take the partial derivatives. However, the set of equations obtained by taking derivatives and setting them to zero is not solvable algebraically because they are not linear with respect to *B*s. A numerical method of iteration known as Newton-Raphson method can be used to solve the set of probability equations.

$$qi(event|Xi) = \frac{1}{1+e^{(Bo+B1\ x1i+B2\ x2i+\ldots+Bk\ xki)}}$$

EVALUATION

In order to evaluate the two methods described in the previous sections, data was extracted from a trauma registry database comprised of information on all trauma patients admitted to a Level I Trauma Center. Two-thousand-one-hundred-fifty-five records, representing all trauma patients admitted in 1992 for more than 24 hours or who died in the Emergency Department, were grouped into two databases as follows: (1) discharge status of "died" (containing 151 records), and (2) any discharge status other than "died" (containing 2004 records). Both databases contained the same variables:

age; initial temperature; Glasgow Coma Scale (GCS) along with its three component subscores (eye opening, verbal response, motor response); initial Trauma Score (TS) in the Emergency Department; coded description of safety equipment in use by the victim at the time of the trauma; coded description of airway management procedures used; coded results of peritoneal lavage, abdominal CT scan, head CT scan, and angiogram; alcohol level; initial hematocrit; concurrent medical history (grouped into one of 10 general categories); Injury Severity Score (ISS); and whether drug screens were done or not and, if so, whether sedatives, opiates, cocaine, marijuana, and/or benzodiazepines were present. The two databases were then linked into a single dataset containing 2155 records, of which the first 151 patients died and the remainder survived.

To generate a systematic selection of patient records to be used for the development of prediction models, each record number not evenly divisible by 10 was placed into a training database containing 1940 records. Information regarding patient discharge status in the training set was used by the investigators to generate their prediction models.

The records of the remaining 10% of patients (those whose record numbers were evenly divisible by 10) were placed into a test database containing 215 records. The discharge status variable of these records was deleted, so the investigators could test the efficacy of the two prediction methods while blinded to the patient's outcome. The investigators were also blinded to the record selection criteria.

The SAS software package was used to perform the logistic regression procedure and determine the coefficients. After the initial run, interactions that take into account products of two variables were added to the model. The model was run in stepwise manner that only entered variables that added a significant improvement in predicting the probability of surviving. Table 1 contains the coefficients for those variables that the logistic determined to be of value in predicting survival using the training set. Then, using these coefficients and the independent variables for each person in the test set, a probability of survival for each individual in the test set was calculated. If the probability was determined to be greater than 50%, it predicted that they had survived, else that they had died.

TABLE 1. Logistic Regression Coefficients

Variable	Coef.	P-value
intercept	-3.2725	
age between 21 and 40	1.3525	0.0002
eye opening = 2 (Glasgow Coma scale)	1.5342	0.0256
eye opening = 3	-1.7873	0.0042
initial trauma score	0.5881	0.0001
airway management (EMS intervention)	-1.5564	0.0007
airway management (ER intervention)	-2.5203	0.0001
head CT scan=positive)	0.9459	0.0079
head CT scan=negative	2.939	0.0001
cardiac complications	-0.9126	0.0196
interaction between Injury Severity Score and Initial Trauma Score	-0.00785	0.0001

INC2.5, an artificial intelligence classification tool, was run on SUN SPARC workstation using C/C++ language. Out of the training data set provided to the investigators, a new subset was created by pulling in the records of all patients who died and an equal number of randomly selected alive patients. We used this pool of records to generate a learning curve by creating decision trees of size 10, 20, ..., 200 and subsequently evaluated their predictive ability by testing the obtained decision tress on the set-aside subset of 70 records. For each run the training and testing sets were randomly selected. A tree of size 100 proved to be the best "performer." This tree was then used for the final evaluation of the method.

The results of the predictive evaluation of the two methods are summarized in Table 2 (LR stands for logistic regression and CF stands for concept formation).

TABLE 2 Predictive Evaluation Results

Predicted Outcome	Actual Outcome	
	Lived	*Died*
Lived (LR)	198	4
Died (LR)	2	11
Lived (CF)	185	2
Died (CF)	15	13
Total	200	15

The sensitivity and specificity information is given in Table 3.

TABLE 3 Sensitivity and Specificity Results

Method	Sensit.	Specific.
Logistic Regression	0.733	0.99
Concept Formation	0.867	0.925

DISCUSSION

Tables 2 and 3 reveal mixed results. Although both systems performed reasonably well, Logistic Regression achieved better performance in terms of specificity, whereas Concept Formation scored better on the sensitivity measure. In addition, Concept Formation had a lower standard deviation with respect to those two measures. Due to INC2.5's ability to build decision trees in an automated fashion, unlike logistic regression which requires some manual tuning, the authors did not do any direct comparison of time required to build models.

In general, Concept Formation offers a more flexible method of dealing with situations where noisy data is expected to be classified in three or more outcome classes. This method not only generates a decision tree, but it also provides a summary description of each intermediate node formed in the decision tree. Such information can be useful in situations where an explanation of the system recommendation is likely to be requested by the user.

In conclusion, it is clear that each system has something to offer to the medical community. More specifically, Logistic Regression provides a well defined formal method of analyzing complex classification domains, while Concept Formation adds flexibility and ability to successfully cope with incorrect and incomplete data. Consequently, we are hoping to look into the issues of combining the two methods into a single tool in the near future.

REFERENCES

AGRESTI, A. (1990), *Categorical Data Analysis.* John Wiley & Sons. New York.

BOHREN, B.F. and HADZIKADIC, M. (1994), Turning Medical Data into Decision-Support Knowledge. *Proceedings of the 18th SCAMC*, 735-739.

CORNFIELD, J. (1962), Joint Dependence of Risk of Coronary Heart Disease on Serum Cholesterol and Systolic Blood Pressure: A Discriminant Function Analysis. *Federation Proceedings*, supplement II, 58-61.

COX, D.R. (1970), *The Analysis of Binary Data.* Methuen, London.

HADZIKADIC, M. (1992a), Automated Design of Diagnostic Systems. *Artificial Intelligence in Medicine Journal*, **4**, 329-342.

HADZIKADIC, M. (1992b), Prediction Performance as a Function of the Representation Language in Concept Formation Systems. *Proceedings of the 14th Annual Conference of the Cognitive Science Society*, 850-854.

HADZIKADIC, M. (1991), Context-Sensitive, Distributed, Variable-Representation Category Formation. *Proceedings of the 13th Annual Meeting of the Cognitive Science Society*, 269-274.

HADZIKADIC, M. and YUN, D. Y. Y. (1989), Concept Formation by Incremental Conceptual Clustering. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, 831-836.

TVERSKY, A. (1977), Features of Similarity. *Psychological review*, **84**, 327-352.